

To: Distribution

Date: Jan. 8, 1985

Cc: Adam Chowaniec

From: Martin Przybylski *Martin*

Subject: Software Operating Requirements, REV. 1

This is the first revision of a memo on Software Operating Requirements dated Jan 2, 1985.

The software operating requirements is a high level statement of the minimal set of operating features for our machine. The purpose of distributing this summary is to aid in our design decisions and tradeoffs. Decisions must be made to ensure that no feature that exceeds these software requirements, and increases the likelihood of missing any milestone, should be included in the software.

Please review this summary again, and return it with your suggestions for additions or corrections to me (at Amiga).

Distribution: Rick Geiger
All Software Engineers

Software Operating Requirements

- 1) The user will be able to point to an application data file, open it, and have the corresponding application running with that specific data file loaded.
- 2) The user will be able to change the default application associated with any data file (when the links are established it will be assumed that the new application can interpret the data file).
- 3) There will be a print spooler capable of running as a background process. The associated print spool file will be on disk and that disk will be loaded (no attempt will be made to prevent disk thrashing if the user begins to use a disk not containing the spool file).
- 4) The machine will have the capability to run multiple processes, giving the user flexibility in choosing his unique foreground process. When multiple processes are running some of the system resources are single threaded (e.g. disk, keyboard, joystick ports), while some are nonsharable (e.g. serial port, parallel port).
- 5) The priority of background processes will be dependent on the user and will be dynamically reassignable.
- 6) The floppy disk is always maintained in a consistent state if deselected (i.e. the light is off). Both applications and the user can flush the buffers on command. The disk motor will be turned off when the disk is not in use.
- 7) The OS has to be able to identify volumes.
- 8) Exchanging floppy disks automatically mounts the new volume. A volume check (e.g. UNIX equivalent of fsck) will be done on command from the user.
- 9) There will be a utility to correct corrupt file systems with minimal user interaction.
- 10) The machine does not recall any data or file names on removed volumes; soft links can exist across volumes.

- 11) User system configuration data will be maintained on each bootable disk, except write protected application media, in a unique file. Only the configuration data on the booted disk will be interpreted by the system. This file will indicate whether the canvas should use the 60 or 80 column fonts, and other peripheral information.
- 12) A help facility will always be available to the user. Each application must be capable of responding with information on what is going on in its canvas when the help key is used.
- 13) The user may at anytime the FDD is deselected remove his floppy disk and switch off the machine.
- 14) Removal of files and other major status changes cannot occur by accidental operation of the mouse or keyboard -- i.e. at least two operations are required to remove a file, such as first moving it into a trash can.
- 15) Where possible the user will be able to undo his/her last command.
- 16) Most applications will come up taking full control of a canvas. Special rules (still to be defined) will be followed for applications coming up in a window.
- 17) Applications can overlay themselves; NO interapplication swapping will occur until a hard disk is present in future machines.
- 18) Memory is dynamically allocated to applications at load time; Get Mem and Release Mem system calls will be available to applications; consequently some applications will not run concurrently with others due to RAM space limitations.
- 19) To reduce memory fragmentation some application segments may be doubly indirect to allow the OS to move segments in memory, and maintain a RAM vector table containing pointers to the current locations of segments.
- 20) Copying, moving and deleting files may be accomplished by dragging the file icons with the mouse or using the keyboard for text entry.
- 21) All (most??) commands that are mouse driven will also have their keyboard equivalents in a command line processor. The icon based user interface will be a subset of the command line processor user interface.

- 22) The novice user will be familiar enough with the machine to do useful things with twenty minutes of training.
- 23) User ID's will not be maintained on any single user machine.
- 24) A time of day and date feature will allow time stamping of files and events. The time stamp will have to be reset each time the machine is rebooted, and on a cold start.
- 25) Nice applications will follow a specific set of rules; e.g. using only 75% of the available memory to allow multiprocessing, release memory when it is no longer needed, follow the voluntary file locking conventions, ...

**brought to you by
andy finkel**